Text Recognition in Videos using a Recurrent Connectionist Approach

Khaoula Elagouni[1,2], Christophe Garcia[3], Franck Mamalet[1], and Pascale Sébillot[2]

[1]Orange Labs R&D, 35512 Cesson Sévigné, France
khaoula.elagouni@orange.com,franck.mamalet@orange.com
[2]IRISA, INSA de Rennes, 35042 Rennes, France
pascale.sebillot@irisa.fr
[3]LIRIS, INSA de Lyon, 69621 Villeurbane, France
christophe.garcia@liris.cnrs.fr

Abstract. Most OCR (Optical Character Recognition) systems developed to recognize texts embedded in multimedia documents segment the text into characters before recognizing them. In this paper, we propose a novel approach able to avoid any explicit character segmentation. Using a multi-scale scanning scheme, texts extracted from videos are first represented by sequences of learnt features. Obtained representations are then used to feed a connectionist recurrent model specifically designed to take into account dependencies between successive learnt features and to recognize texts. The proposed video OCR evaluated on a database of TV news videos achieves very high recognition rates. Experiments also demonstrate that, for our recognition task, learnt feature representations perform better than hand-crafted features.

Keywords: Video text recognition, multi-scale image scanning, ConvNet, LSTM, CTC.

1 Introduction

Visual patterns in multimedia documents usually contain relevant information that allows content indexing. In particular, texts embedded in videos often provide high level semantic clues that can be used to develop several applications and services such as multimedia documents indexing and retrieval schemes, teaching videos and robotic vision systems. In this context, the design of efficient Optical Character Recognition (OCR) systems specifically adapted to video data is an important issue. However, the huge diversity of texts and their difficult acquisition conditions (low resolution, complex background, non uniform lighting, etc.) make the task of video embedded text recognition a challenging problem.

Most of prior research in OCR has focused on scanned documents and handwritten text recognition. Recently, systems dedicated to embedded texts in video have generated a significant interest in the OCR community [10, 2, 12]. Most of the proposed approaches rely on an initial segmentation step that splits texts

into individual characters (a complete survey of character segmentation methods is presented in [1]) and a second one that recognizes each segmented character. However, the different distortions in text images and the low resolution of videos make the segmentation very hard, leading to poor recognition results. To improve performance, some authors reduce segmentation ambiguities by considering character recognition results and by introducing some linguistic knowledge [4]. In our recent work [3] dedicated to natural scene text recognition, an approach that avoids the segmentation step by using a multi-scale character recognition and a graph model was proposed. Even though this method obtained good results, its main drawback remains the high complexity of the graph model which has to deal with all the recognition results. Other recent work applied to handwriting recognition [7] has also proposed to avoid any segmentation, using a connectionist model that relies on a recurrent neural network (RNN) trained with hand-crafted features extracted from the input image. In this paper, we adapt this idea of absence of segmentation step to the task of video text recognition, and propose an OCR scheme that relies on a connectionist temporal approach; a second contribution lies in a multi-scale representation of text images by means of learnt features particularly robust to complex background and low resolution.

The remainder of the paper is organized as follows: after presenting the outlines of our approach in section 2, we detail our method to generate feature-based representations of texts (section 3). Section 4 introduces the fundamentals of the recurrent neural model used and describes the chosen architecture. Finally, experiments and obtained results are reported in section 5 before concluding and highlighting our future work in section 6.

2 Proposed Approach

The first task for video text recognition consists in detecting and extracting texts from videos as described in [4]. Once extracted, text images are recognized by means of two main steps as depicted in fig. 1: generation of text image representations and text recognition. In the first step, images are scanned at different scales so that, for each position in the image, four different windows are extracted. Each window is then represented by a vector of features learnt with a convolutional neural network (ConvNet). Considering the different positions in the scanning step and the four windows extracted each time, a sequence of learnt features vectors $\left[X^0, \ldots, X^t, \ldots, X^p\right]$ is thus generated to represent each image. The second step of the proposed OCR is similar to the model presented in [7], using a specific bidirectional recurrent neural network (BLSTM) able to learn to recognize text making use of both future and past context. The recurrent network is also characterized by a specific objective function (CTC) [7], that allows the classification of non-segmented characters. Finally, the network's outputs are decoded to obtain the recognized text. The following sections describe these different steps and their interactions within the recognition scheme.

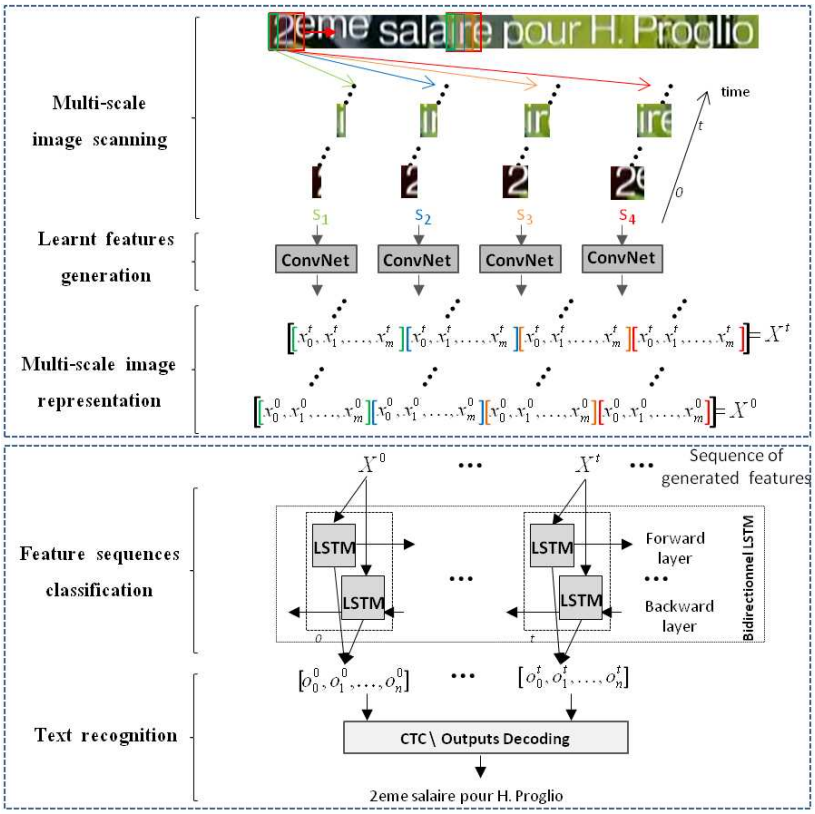

Fig. 1. Scheme of the proposed approach.

3 Multi-Scale Feature Learning for Character Representation

Our objective is to produce a relevant representation of texts extracted from videos, which has to be robust to noise, deformations, and translations. To that end, text images are scanned with different window sizes (*cf.* section 3.1) then each window is represented by a set of learnt features (*cf.* section 3.2).

3.1 Multi-scale Image Scanning Scheme

Text images usually consist of a succession of characters having different sizes and shapes depending on their labels. We therefore propose to scan each full text image at several scales and at regular and close positions (typically, a step of $\frac{h}{8}$, where h is the image height) to ensure that at least one window will be aligned with each character in the image. Thus, at each position, different scales are considered to handle various character sizes. Experiments have shown that good results are obtained with four windows of widths $\frac{h}{4}$, $\frac{h}{2}$, $\frac{3h}{4}$ and h.

Furthermore, since the characters can have different morphologies, we adapt the window borders to the local morphology of the image and hence, when possible, clean the neighborhood of characters. For each window position and scale, the borders within the full image are computed as follows. (For figure clearness the computation of non linear border is not shown in fig. 1, but interested readers can refer to [4] to get more details) Assuming that pixels in text images belong to two classes—"text" and "background"—, a pre-processing step generates a fuzzy map which encodes, for each pixel, its membership degree to class "text". Using a shortest path algorithm within the obtained map, non-linear vertical borders are computed, following pixels that have a low probability to belong to the class "text". In case of complex background or non separated characters, the shortest path algorithm induces straight vertical borders.

3.2 Neural-based Model for Features Learning

For each position in the text, considering different scales, four windows are extracted, from which accurate representations that preserve the information useful for the recognition task have to be found. In [7], Graves *et al.* have used hand-crafted features which are known not to be robust to noise, deformations. We thus propose to use learnt features. In this context, Convolutional Neural Networks (ConvNets) [9] have shown to be well adapted [11] and particularly robust to complex background and low resolution. A ConvNet is a bio-inspired hierarchical multi-layered neural network able to learn visual patterns directly from the image pixels without any pre-processing. Relying on specific properties (local receptive fields, weight sharing and sub-sampling), this model learns to extract appropriate descriptors and to recognize characters at the same time.

The proposed method consists in representing sliding windows by the descriptors learnt by ConvNets. First a ConvNet is trained to classify images of individual characters. Then, the stimulation of this network is applied on each window (being a character or not), and the vector of the penultimate layer activations, considered as a feature extraction layer, is used as the window's descriptor. In our experiments, several configurations of ConvNets have been tested. The best configuration takes as input a color window image mapped into three 36×36 input maps, containing values normalized between -1 and 1, and returns a vector of values normalized with the softmax function. The architecture of our ConvNet is similar to the one presented in [4] and consists of six hidden layers. The first four ones are alternated convolutional and sub-sampling layers connected to three other neuron layers where the penultimate layer contains 50 neurons. Therefore, using this network architecture, each position in the text image is represented by a vector of 200 values (50 values for each scale window) corresponding to the features learnt by the ConvNet model.

4 Text Recognition using a Recurrent Neural Model

Once text images are represented by sequences of automatically learnt features, we combine a particular RNN (BLSTM) and a connectionist classification model

(CTC) to build a model able to learn how to classify the feature sequences and hence recognize texts. While the BLSTM allows to handle long-range dependencies between features, the CTC enables our scheme to avoid any explicit segmentation in characters, and learn to recognize jointly a sequence of classes and their positions in the input data.

4.1 Bidirectional Long Short-Term Memory (BLSTM)

The basic idea of RNN is to introduce recurrent connections which enable the network to maintain an internal state and thus to take into account the past context. However, these models have a limited "memory" and are not able to look far back into the past [8] becoming insufficient when dealing with long input sequences, such as our feature sequences. To overcome this problem, the Long Short-Term Memory (LSTM) [5] model was proposed to handle data with long range interdependencies. A LSTM neuron contains a constant "memory cell"—namely constant error carousel (CEC)—whose access is controlled by some multiplicative gates. For these reasons we chose to use the LSTM model to classify our learnt feature sequences. Moreover, in our task of text recognition, the past context is as important as the future one (*i.e.*, both previous and next letters are important to recognize the current letter). Hence, we propose to use a bidirectional LSTM which consists of two separated hidden layers of LSTM neurons. The first one permits to process the forward pass making use of the past context, while the second serves for the backward pass making use of the future context. Both hidden layers are connected to the same output layer (*cf.* fig. 1).

4.2 Connectionist Temporal Classification (CTC)

Even though BLSTM networks are able to model long-range dependencies, as for classical RNNs, they require pre-segmented training data to provide the correct target at each timestep. The Connectionist Temporal Classification (CTC) is a particular objective function defined [6] to extend the use of RNNs to the case of non-segmented data. Given an input sequence, it allows the network to jointly learn a sequence of labels and their positions in the input data. By considering an additional class called "Blank", the CTC enables to transform the BLSTM network outputs into a conditional probability distribution over label sequences ("Blank" and Characters). Once the network is trained, CTC activation outputs can be decoded, removing the "Blank" timesteps, to obtain a sequence of labels corresponding to a given input sequence. In our application, a best path decoding algorithm is used to identify the most probable sequence of labels.

4.3 Network Architecture and Training

After testing several architectures, a BLSTM with two hidden layers of 150 neurons, each one containing recurrent connexions with all the other LSTM cells and fully connected to the input and the output layers, has been chosen. The

network takes as input a sequence of vectors of 200 values normalized between -1 and 1 and returns a sequence of vectors of 43 outputs (42 neurons corresponding to 42 classes of characters, namely letters, numbers and punctuation marks, and one more neuron for the class "Blank"). In our experimental data, depending on the text image size, the sequence of inputs can contain up to 300 vectors. The network is trained with the classical back-propagation through time algorithm using a learning rate of 10^{-4} and a momentum of 0.9.

5 Experimental Setup and Results

This section reports several tests and discusses obtained results. After presenting the datasets, the proposed OCR scheme is evaluated and compared to other state-of-the-art methods. Learnt feature representations are shown to outperform hand-crafted features, leading to better recognition results.

5.1 Datasets

Our experiments have been carried out on a dataset of 32 videos of French news broadcast programs. Each video, encoded by MPEG-4 (H. 264) format at 720×576 resolution, is about 30 minutes long and contains around 400 words which correspond to a set of 2200 characters (*i.e.*, small and capital letters, numbers and punctuation marks). Embedded texts can vary a lot in terms of size (from 8 to 24 pixels of height), color, font and background. Four videos were used to generate a dataset of 15168 images of single characters perfectly segmented. This database—called *CharDb*—consists of 42 classes of characters (26 letters, 10 numbers, the space character and 5 special characters; namely '.', '-', '(', ')' and ':') and is used to train the ConvNet described in section 3.2. The remaining videos were annotated and divided into two sets: *VidTrainDb* and *VidTestDb* containing respectively 20 and 8 videos. While the first one is used to train the BLSTM, the second is used to test the complete OCR scheme.

5.2 Experimental Results

The training phase of the BLSTM is performed on a set of 1399 text images extracted from *VidTrainDb*. Once trained, the BLSTM network is evaluated on a set of 734 text images extracted from *VidTestDb*. To evaluate the contribution of learnt features, the BLSTM was trained with two types of input features (namely hand-crafted and learnt ones) and evaluated with the Levenshtein distance.

On the one hand, texts are represented by sequences of hand-crafted features as proposed in [7] for handwriting recognition. In this experimentation, text images are first binarized; then nine geometrical features per column are extracted (average, gravity center, etc.). The BLSTM trained with these features achieves a good performance of 92.73% of character recognition rate (*cf.* table 1).

On the other hand, as proposed in section 3, we represent text images by means of multi-scale learnt features. For this, the ConvNet was trained to recognize image of characters on 90% of *CharDb*, and its classification performance

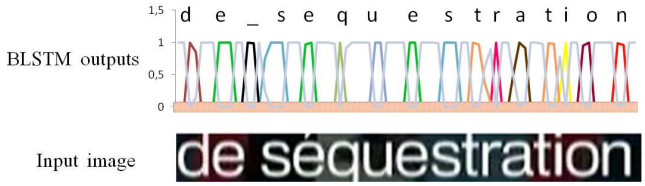

Fig. 2. Example of recognized text: each class is represented by a color, the label "_" represents the class "space" and the gray curve corresponds to the class "Blank".

Used features	Character recognition rate
Geometrical features	92.73%
Learnt features	97.18%

Table 1. Usefulness of learnt features.

was evaluated on the remaining 10%. A very high recognition rate of 98.04% was obtained. Learnt features were thus generated with the trained ConvNet and used to feed the BLSTM. Fig. 2 illustrates an example of recognized text and shows its corresponding BLSTM outputs where each recognized character is represented with a peak. Even though extracted geometrical features achieve good performance, for our application, they seem to be less adapted than learnt features which obtain a high character recognition rate of 97.18% (*cf.* table 1). The main improvement is observed for text images with complex background, for which the geometrical features introduced high inter-class confusions.

We further compare our complete OCR scheme to another previously published method [4] and commercial OCR engines; namely ABBYY, tesseract, GNU OCR, and SimpleOCR. Using the detection and extraction modules proposed in [4], these different systems were tested and their performances were evaluated. As shown in table 2, the proposed OCR yields the best results and outperforms commercial OCRs.

6 Conclusions

We have presented an OCR scheme adapted to the recognition of texts extracted from digital videos. Using a multi-scale scanning scheme, a novel representation of text images built with features learnt by a ConvNet is generated. Based on a particular recurrent neural network—namely the BLSTM—and a connectionist classification—namely the CTC—our approach takes as input generated representations and recognizes texts. Besides its ability to make use of learnt feature dependencies, the proposed method permits to avoid the difficult character segmentation step. Our experiments have highlighted that learnt feature representations are well-adapted to texts embedded in videos and yield to better results than hand-crafted features. Our complete scheme was evaluated on a dataset of

Used features	Character recognition rate	Word recognition rate
Proposed OCR scheme	97.35%	87.20%
Elagouni et al. [4]	94.95%	78.24%
ABBYY engine	94.68%	87.23%
Tesseract engine	88.19%	69.22%
SimpleOCR engine	73.58%	29.01%
GNU OCR engine	56.47%	12.79%

Table 2. Comparison of the proposed scheme to a state-of-the-art method and commercial OCR engines.

news TV videos and obtained promising results (exceeding 97% of characters and 87% of words correctly recognized) outperforming other state-of-the-art methods and commercial OCRs. As future extensions of this work, we plan to test our approach on scene text images (*i.e.*, no longer on embedded text) and also to produce new text representations based on unsupervised learning techniques (autoencoders) and evaluate their contribution to our recognition task.

References

1. Casey, R., Lecolinet, E.: A survey of methods and strategies in character segmentation. PAMI 18(7), 690–706 (2002)
2. Chen, D., Odobez, J., Bourlard, H.: Text detection and recognition in images and video frames. PR 37(3), 595–608 (2004)
3. Elagouni, K., Garcia, C., Mamalet, F., Sébillot, P.: Combining multi-scale character recognition and linguistic knowledge for natural scene text OCR. In: DAS. pp. 120–124 (2012)
4. Elagouni, K., Garcia, C., Sébillot, P.: A comprehensive neural-based approach for text recognition in videos using natural language processing. In: ICMR (2011)
5. Gers, F., Schraudolph, N., Schmidhuber, J.: Learning precise timing with lstm recurrent networks. JMLR 3(1), 115–143 (2003)
6. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In: ICML. pp. 369–376 (2006)
7. Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., Schmidhuber, J.: A novel connectionist system for unconstrained handwriting recognition. PAMI 31(5), 855–868 (2009)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation 9(8), MIT Press (1997)
9. LeCun, Y., Bengio, Y.: Convolutional networks for images, speech, and time series. The Handbook of Brain Theory and Neural Networks. MIT Press (1995)
10. Lienhart, R., Effelsberg, W.: Automatic text segmentation and text recognition for video indexing. Multimedia Systems 8(1), 69–81 (2000)
11. Saidane, Z., Garcia, C.: Automatic scene text recognition using a convolutional neural network. In: ICBDAR. pp. 100–106 (2007)
12. Yi, J., Peng, Y., Xiao, J.: Using multiple frame integration for the text recognition of video. In: ICDAR. pp. 71–75 (2009)